**校园社区平台**

作者姓名：赵安玉

专业名称：软件工程

指导教师：张 玺 助教

**成都理工大学工程技术学院**

**学位论文诚信承诺书**

**本人慎重承诺和声明：**

1.本人已认真学习《学位论文作假行为处理办法》（中华人民共和国教育部第34号令）、《成都理工大学工程技术学院学位论文作假行为处理实施细则（试行）》（成理工教发〔2013〕30号）文件并已知晓教育部、学院对论文作假行为处理的有关规定，知晓论文作假可能导致作假者被取消学位申请资格、注销学位证书、开除学籍甚至被追究法律责任等后果。

2.本人已认真学习《成都理工大学工程技术学院毕业设计指导手册》，已知晓学院对论文撰写的内容和格式要求。

3.本人所提交的学位论文（题目：网上拍卖系统），是在指导教师指导下独立完成，本人对该论文的真实性、原创性负责。若论文按有关程序调查后被认定存在作假行为，本人自行承担相应的后果。

承诺人（学生签名）：

20 年 月 日

注：学位论文指向我校申请学士学位所提交的本科学生毕业

**摘要**

图书馆为高校师生的日常生活、工作提供了一个很好的交流学习平台。现如今，科技高速发展伴随着生活的日新月异，有许多知识需要我们不断的汲取，以便使自身发展能够跟上社会发展的脚步。而大学这个特殊的时期正是我们广泛学习知识，扩展自己眼界的一个重要时期，所以图书管理系统的研究开发很重要，有利于高校对图书的合理有效利用。

图书管理系统，在实际生活中主要用于管理各机关单位以及各高校的书籍。图书馆管理人员借助图书管理系统能更高效有序的对图书馆的书藉进行管理，书籍的有序管理在一定程度上也有助于用户更高效率的使用书籍，提高书籍的使用效率。图书管理系统，借助网上平台的优势，在使用者输入其独有的用户名和密码后方可进入系统，在使用者完成书籍借阅和书籍归还等结束操作后，可退出系统。但使用者在图书管理系统的使用过程中需要遵循一定的规则，即使用者须在借阅逾期前还书。另外，使用者可查看书籍借阅榜，即读者排行榜和热门书籍排行榜。

本系统主要是采用Servlet技术作为后台控制器，控制后台数据的处理，选用Jdbc+dbutils来对后台数据库进行连接并对数据库中的表进行增加、删除、修改、查询等操作，选用EasyUI作为前端框架，jQuery可提供很多Ajax的方法，便于高效的请求数据。在本文的最后，通过学习一些现在广泛存在的一些测试方法对该系统进行测试，测试结果显示，各功能模块所实现的基本功能符合图书管理系统的日常需求，基本达到了图书管理系统预期设计目标。

**关键词**：图书管理系统 JavaWeb Jsp Servlet

**Abstract**

The library provides a good platform for the daily life and work of college teachers and students. Nowadays, with the rapid development of science and technology, with the rapid change of life, there is a lot of knowledge we need to constantly learn, so that our own development can keep up with the pace of social development. The special period of university is an important period for us to study knowledge extensively and expand our vision, so the research and development of library management system is very important, which is conducive to the reasonable and effective use of books in Colleges and universities.

The book management system is mainly

目录

[前言](#_Toc1241382779_WPSOffice_Level1) [1](#_Toc1241382779_WPSOffice_Level1)

[1 绪论](#_Toc1924842408_WPSOffice_Level1) [2](#_Toc1924842408_WPSOffice_Level1)

[2 校园社区平台技术概述](#_Toc1985572816_WPSOffice_Level1) [3](#_Toc1985572816_WPSOffice_Level1)

[2.1 概念](#_Toc1924842408_WPSOffice_Level2) [3](#_Toc1924842408_WPSOffice_Level2)

[2.1.1 基本概念](#_Toc1924842408_WPSOffice_Level3) [3](#_Toc1924842408_WPSOffice_Level3)

[1. J2EE的概念](#_Toc1985572816_WPSOffice_Level3) [3](#_Toc1985572816_WPSOffice_Level3)

[2. 模块化设计的概念和思想](#_Toc228347307_WPSOffice_Level3) [3](#_Toc228347307_WPSOffice_Level3)

[3. 前后端分离架构技术概念](#_Toc1939882815_WPSOffice_Level3) [3](#_Toc1939882815_WPSOffice_Level3)

[4. MVC 设计模式概念](#_Toc1934985626_WPSOffice_Level3) [3](#_Toc1934985626_WPSOffice_Level3)

[5. Spring 框架](#_Toc1658063788_WPSOffice_Level3) [3](#_Toc1658063788_WPSOffice_Level3)

[6. SpingBoot 概念](#_Toc939219241_WPSOffice_Level3) [3](#_Toc939219241_WPSOffice_Level3)

[7. MybatisPlus 框架概念](#_Toc82964735_WPSOffice_Level3) [3](#_Toc82964735_WPSOffice_Level3)

[8. GraphQL API 查询语言的概念](#_Toc2031322322_WPSOffice_Level3) [3](#_Toc2031322322_WPSOffice_Level3)

[9. JWT (Java Web Token)](#_Toc1657395528_WPSOffice_Level3) [3](#_Toc1657395528_WPSOffice_Level3)

[10. VUE 框架的概念](#_Toc1347507790_WPSOffice_Level3) [3](#_Toc1347507790_WPSOffice_Level3)

[3 平台架构设计](#_Toc228347307_WPSOffice_Level1) [4](#_Toc228347307_WPSOffice_Level1)

[3.1 开发技术及版本依赖](#_Toc1985572816_WPSOffice_Level2) [4](#_Toc1985572816_WPSOffice_Level2)

[3.1.1 开发环境](#_Toc303341540_WPSOffice_Level3) [4](#_Toc303341540_WPSOffice_Level3)

[3.1.2 开发技术](#_Toc256473354_WPSOffice_Level3) [4](#_Toc256473354_WPSOffice_Level3)

[3.2 平台架构](#_Toc228347307_WPSOffice_Level2) [4](#_Toc228347307_WPSOffice_Level2)

[3.2.1 平台后端模块概述](#_Toc1176623100_WPSOffice_Level3) [4](#_Toc1176623100_WPSOffice_Level3)

[4 平台详细设计及实现](#_Toc1939882815_WPSOffice_Level1) [6](#_Toc1939882815_WPSOffice_Level1)

[4.1 平台数据库设计](#_Toc1939882815_WPSOffice_Level2) [6](#_Toc1939882815_WPSOffice_Level2)

[4.1.1 数据库表设计](#_Toc2119346457_WPSOffice_Level3) [6](#_Toc2119346457_WPSOffice_Level3)

[1. 用户表 user](#_Toc693853276_WPSOffice_Level3) [6](#_Toc693853276_WPSOffice_Level3)

[2. 评论表 comment](#_Toc243831082_WPSOffice_Level3) [7](#_Toc243831082_WPSOffice_Level3)

[3. 帖子表 post](#_Toc2002858936_WPSOffice_Level3) [7](#_Toc2002858936_WPSOffice_Level3)

[4. 消息表 message](#_Toc310922559_WPSOffice_Level3) [8](#_Toc310922559_WPSOffice_Level3)

[4.2 平台后端设计](#_Toc1934985626_WPSOffice_Level2) [8](#_Toc1934985626_WPSOffice_Level2)

[4.2.1 持久层设计](#_Toc533861262_WPSOffice_Level3) [8](#_Toc533861262_WPSOffice_Level3)

[5 测试与分析](#_Toc131830690_WPSOffice_Level3) [9](#_Toc131830690_WPSOffice_Level3)

[总结](#_Toc1934985626_WPSOffice_Level1) [10](#_Toc1934985626_WPSOffice_Level1)

[参考文献](#_Toc1658063788_WPSOffice_Level1) [11](#_Toc1658063788_WPSOffice_Level1)

**前言**

**1 绪论**

# 2 校园社区平台技术概述

**2.1 概念**

**2.1.1 基本概念**

1. J2EE的概念

J2EE安全技术是一种通过使用了Java的平台和数据库来有效简化与企业和公司系统解决方案的设计和开发,增长和有效管理与业务有关的复杂安全问题的系统结构。

1. 模块化设计的概念和思想

模块化[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1/223952" \t "/home/anyu/Documents\\x/_blank)是指在进行[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1" \t "/home/anyu/Documents\\x/_blank)时将一个大程序按照功能划分为若干小程序模块，每个小程序模块完成一个确定的功能，并在这些模块之间建立必要的联系，通过模块的互相协作完成整个功能的程序设计方法。在设计较复杂的程序时，一般采用自顶向下的方法，将问题划分为几个部分，各个部分再进行细化，直到分解为较好解决问题为止。模块化的目的是为了降低程序复杂度，使程序设计、调试和维护等操作简单化。

1. 前后端分离架构技术概念
2. MVC 设计模式概念
3. Spring 框架

Spring框架是[Java平台](https://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0/3793459" \t "/home/anyu/Documents\\x/_blank)上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为[EJB](https://baike.baidu.com/item/EJB/144195" \t "/home/anyu/Documents\\x/_blank)（EnterpriseJavaBeans）模型的补充，甚至是替补。它简化了 Java 企业级应用开发。

1. SpingBoot 概念
2. MybatisPlus 框架概念
3. GraphQL API 查询语言的概念
4. JWT (Java Web Token)
5. VUE 框架的概念

# 3 平台架构设计

**3.1 开发技术及版本依赖**

本节主主要介绍校园社区平台所用到的技术和开发环境，以及所依赖其他资源。

**3.1.1 开发环境**

操作系统：Linux 510

开发工具：IDEA 2020.3 Navicat

数据库：MySQL 8.0.20 Redis

容器技术：Docker

测试工具：Postman

SpringBoot 2.3.4.RELEASE

MybatisPlus 3.4.0

Java 11

Gradle 6.7

Vue 2

**3.1.2 开发技术**

**3.2 平台架构**

平台是一个前后端分离的web应用。后端使用了Gradle作为版本控制工具和模块管理工具；基于SpringBoot 的框架做快速开发; MybatisPlus做持久层框架,进行数据库操作。前端使用Vue框架搭建前端应用，利用Element 进行快速页面搭建。使用GraphQL作为前后端通信方式。

**3.2.1 平台后端模块概述**

各个模块遵循软件开发模块化设计原则，每个模块都只关注自己的功能或业务。其中ancf-msg、ancf-user、ancf-post业务模块按MVC模式实现；理论上 ancf-auth、ancf-oss、ancf-cache对外部模块只暴露业务接口。

1. ancf-webapp 模块，它是程序的主入口模块，最顶层模块；直接依赖于ancf-msg、ancf-user、ancf-post业务模块。程序的配置文件存放该模块中。

2. ancf-msg 模块，它是校园社区平台用户间消息发送、平台消息推送的业务实现。

3. ancf-user 模块，它是校园社区平台与用户相关业务的实现，包括用户注册、用户登录、头像修改等等。

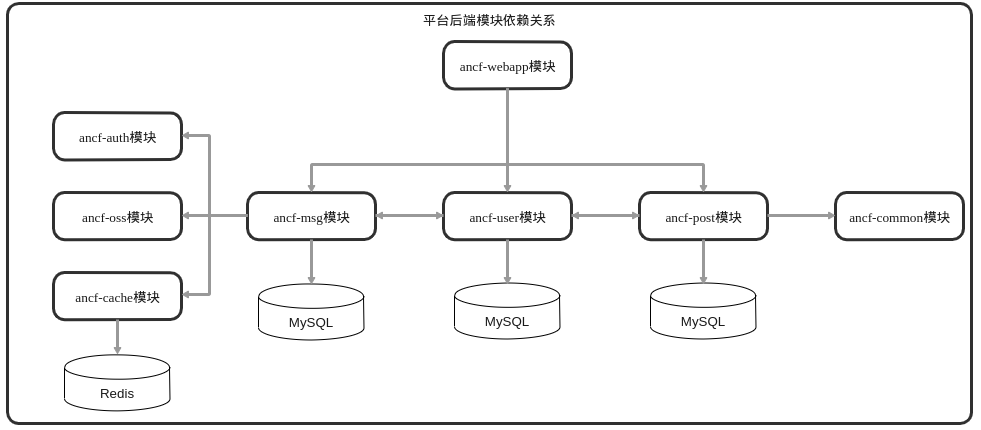
4. ancf-post 模块，它是校园社区论坛与帖子相关业务的实现，包括发帖，评论、点赞等等。

5. ancf-common 模块，它是公共模块，最底层模块，基本上所有的模块都依赖于它。

6. ancf-auth 模块，它是平台权限认证、安全管理的实现。

7. ancf-oss模块，它是平台文件上传云端服务器的实现，借助于阿里云OSS服务。

8. ancf-cache 模块，它是系统缓存的业务实现。

****

**3.2.2 平台前端概述**

**3.2.3 平台前后端通信技术**

# 4 平台详细设计及实现

**4.1 平台数据库设计**

### 4.1.1 数据库表设计

1. 用户表 user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | ID |
| account | char | 16 | NOT NULL | 账号 |
| gender | tinyint | 1 | NOT NULL | 性别 |
| age | tinyint | 3 | NOT NULL | 年龄 |
| birthday | datetime |  | NOT NULL | 出生日期 |
| nickname | char | 9 | NOT NULL | 昵称 |
| real\_name | char | 6 | NULL | 真实姓名 |
| avatar | char | 100 | NOT NULL | 头像存储地址 |
| email | char | 25 | NULL | 邮箱地址 |
| mobile | char | 11 | NULL | 手机号 |
| salt | char | 5 | NOT NULL | 密码加盐 |
| password | char | 50 | NOT NULL | 密码 |
| activation | tinyint | 1 | NOT NULL | 账号是否激活 |
| status | tinyint | 1 | NOT NULL | 用户状态 |
| create\_time | datetime |  | NOT NULL | 用户创建时间 |
| modified\_time | datetime |  | NOT NULL | 用户修改时间 |

1. 评论表 comment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 评论id |
| content | varchar | 255 | NOT NULL | 内容 |
| user\_id | int | 10 | NOT NULL | 评论者id |
| entity\_id | int | 10 | NOT NULL | 评论的实体id |
| target\_id | int | 10 | NULL | 被评论者的id |
| entity\_type | tinyint | 1 | NOT NULL | 评论实体类型 |
| status | tinyint | 1 | NOT NULL | 评论状态 |
| create\_time | datetime |  | NOT NULL | 评论创建时间 |
| modifid\_time | datetime |  | NOT NULL | 评论修改时间 |

1. 帖子表 post

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 帖子id |
| type | tinyint | 1 | NOT NULL | 帖子类型 |
| user\_id | int | 10 | NOT NULL | 发帖者id |
| title | varchar | 20 | NOT NULL | 帖子标题 |
| content | text |  | NOT NULL | 内容 |
| cmt\_num | int | 10 | NOT NULL | 评论数 |
| score | float |  | NOT NULL | 得分 |
| status | tinyint | 1 | NOT NULL | 帖子状态 |
| create\_time | datetime |  | NOT NULL | 创建时间 |
| modified\_time | datetime |  | NOT NULL | 修改时间 |

1. 消息表 message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 消息id |
| from\_id | int | 10 | NOT NULL | 发送者id |
| to\_id | int | 10 | NOT NULL | 接受者id |
| chart\_id | varchar | 20 | NOT NULL | 聊天id |
| content | varchar | 255 | NOT NULL | 内容 |
| status | tinyint | 1 | NOT NULL | 状态 |
| create\_time | datetime |  | NOT NULL | 创建时间 |
| modified\_time | datetime |  | NOT NULL | 修改时间 |

4.2 平台后端设计

4.2.1 持久层设计

4.2.2

# 5 测试与分析

**总结**

**参考文献**